**Java笔试题**

Part1:<https://wenku.baidu.com/view/de6bae8cee3a87c24028915f804d2b160b4e8621.html>

7.下面关于数组说法正确的是：多选( abcde)

A)一维数组实质上是相同类型变量的列表

B)创建一个数组首先定义数组变量所需的类型

C) char c[]=new char[26];可声明一个含有26个元素的char 型数组

D)当为一个多维数组的时候分配内存时，仅需要为第一指定内存，然后再分配其他维的存

E) int twain[]D] = new int[4][5];可声明一个二维数组

10.若x是float类变量，x=10/4;则x的值是。 (b)

A:2

B:2.0

C:2,5

D:编译错误

12.下面哪种情况能实现自动转换，多选(~~bcd~~，ace)。

A) byte 型转换成int型B) int型转换成byte 型

C) float型转换成double型D) double 型转换成int型E) char型转换成int型

JAVA基本类型之间转换的两种方式：

1. 自动（隐式）类型转换：从小类型到大类型，不需要强制转换符

例如：int a = 5; double b = a

此处将int类型变量a的值赋值给double类型变量b,因为是小类型转换成大类型，int类型变量a的值会自动转换成double类型并赋值给变量b

1. 强制类型转换：从大类型到小类型，需要强制转换符实现强制转换

强制转换符：（需要转换成的类型）变量

double a =5.123; int b=(int)a;.

此处将double类型变量a的值赋值给int类型变量b,因为是大类型转换成小类型，需要强制转换符先将double类型变量a的值转换成int类型再赋值给int类型变量b

1. 关于byte、short、 char赋值与运算的强转规则: .

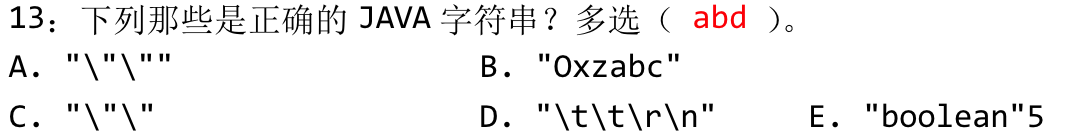
1)不参与运算，整数直接量可以直接赋值给byte, short,char,不需要强转。

2).byte,short,char型变量参与运算时，需要强转

例如: byte b1 = 5;byte b2 =2;byte b3 = (byte)(b1 +b2).

b1与b2不参与运算，整数直接量可以直接赋值给byte;参与运算后，b1 +b2和的值默认

为int类型需要强制转换成byte类型再赋值给byte类型变量b3



String str = **"\"\""**; *//两个\均为转义字符，所以打印不显示，只打印 ""*



14:在使用 super和this关键字时，以下描述正确的是。(a)

A:在子类构造方法中使用super ()显示调用父类的构造方法，super ()必须写在子类构

造方法的第一行，否则编译不通过

B: super ()和this ()不一定要放在构造方法内第一行

C: this ()和super ()可以同时出现在一个构造函数中

D: this ()和super ()可以在static环境中使用，包括static方法和static语句块

Super和this区别：<https://blog.csdn.net/lncsdn_123/article/details/79025525>

* super（参数）：调用基类中的某一个构造函数（应该为构造函数中的第一条语句）
* this（参数）：调用本类中另一种形成的构造函数（应该为构造函数中的第一条语句）
* super:　它引用当前对象的**直接父类**中的成员（用来访问直接父类中被隐藏的父类中成员数据或函数，基类与派生类中有相同成员定义时如：super.变量名 super.成员函数据名（实参）
* this：它代表当前对象名（在程序中易产生二义性之处，应使用this来指明当前对象；如果函数的形参与类中的成员数据同名，这时需用this来指明成员变量名）
* 调用super()必须写在子类构造方法的第一行，否则编译不通过。每个子类构造方法的第一条语句，都是隐含地调用super()，如果父类没有这种形式的构造函数，那么在编译的时候就会报错。
* super()和this()类似,区别是，super()从子类中调用父类的构造方法，this()在同一类内调用其它方法。
* super()和this()均需放在构造方法内第一行。
* 尽管可以用this调用一个构造器，但却不能调用两个。
* this和super不能同时出现在一个构造函数里面，因为this必然会调用其它的构造函数，其它的构造函数必然也会有super语句的存在，所以在同一个构造函数里面有相同的语句，就失去了语句的意义，编译器也不会通过。
* this()和super()都指的是对象，所以，均不可以在static环境中使用。包括：static变量,static方法，static语句块。
* 从本质上讲，this是一个指向本对象的指针, 然而super是一个Java关键字。

15:以下对封装的描述正确的是。(d)

A:只能对一个类中的方法进行封装，不能对属性进行封装

B:如果子类继承了父类，对于父类中进行封装的方法，子类仍然可以直接调用

//只能直接调用非私有的方法。如果是private修饰的方法，子类也是不能调用的

C:封裝的意义不大，因此在编码时尽量不要使用

D:封装的主要作用在于对外隐藏内部实现细节，增强程序的安全性

17:以 下关于final关键字说法错误的是(ac) (两项)

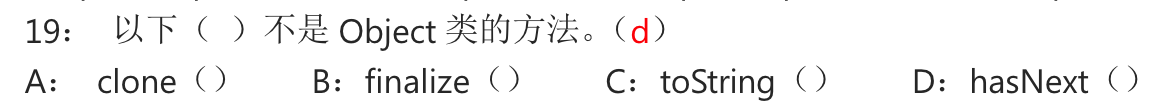
A:final是java中的修饰符，可以修饰类、接口、抽象类、方法和属性

B: final 修饰的类肯定不能被继承

C: final 修饰的方法不能被重载

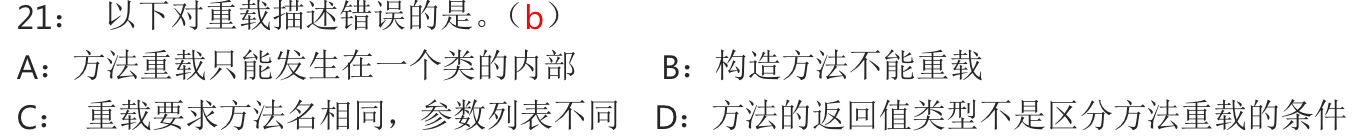
D: final 修饰的变量不允许被再次赋值

final关键字：<https://www.cnblogs.com/xuelisheng/p/11158110.html>





面向对象三大特性：封装、继承、多态



22.以下对接口描述错误的有。(D)

A: 接口没有提供构造方法

// 因为不允许实例化

B: 接口中的方法默认使用public、 abstract修饰

C: 接口中的属性默认使用public、 static、 final修饰

D: 接口不允许多继承

// 说法不严谨，应该是多实现implements

23:以下对目定义异常描述正确的是。(c)

A:自定义异常必须继承Exception

自定义异常通常是定义一个继承自Exception类的子类。一般情况下我们都会直接继承自Exception类，而不会继承某个运行时的异常类

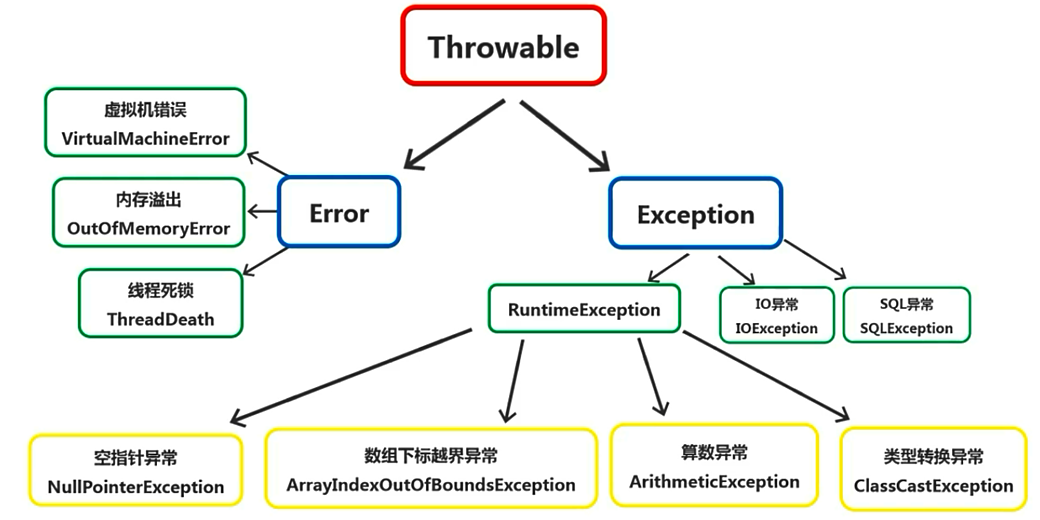
B:自定义异常可以继承自Error

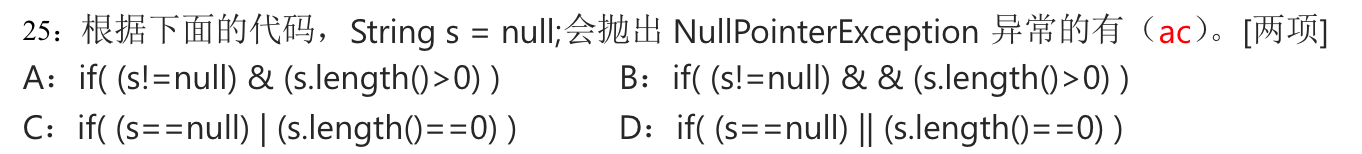
不能，异常是异常，错误是错误

C:自定义异常可以更加明确定位异常出错的位置和给出详细出错信息

D:程序中已经提供了丰富的异常类，使用自定义异常没有意义

Java自定义异常，应该继承Exception还是Runtime Exception：<https://www.zhihu.com/question/51970444>





AC：& |为逻辑运算，而对象是null，所以运算会抛空指针异常。BD为逻辑判断

**27:下列声明哪个是错误的。(b)**

A:inti=10;

B:float f = 1.1;

后面加f ，1.1f

C:double d = 34.4;

D:long m = 4990;

**28:关于抽象类的说法正确的是(b)**

A:抽象类中一定包含抽象方法，否则是错误

B:包含抽象方法的类一定是抽象类

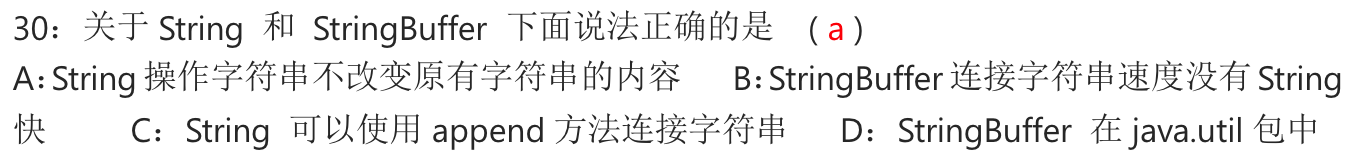
C:抽象方法可以没有方法体，也可以有方法体 //必须没有方法体

D:抽象类的子类一定不是抽象类

抽象类：<https://blog.csdn.net/xkfanhua/article/details/80567557>

抽象类中可以有非抽象方法，含有抽象方法的类一定是抽象类

抽象类不能实例化，需要子类继承才有意义，子类继承后必须实现(重写)其抽象方法



String,StringBuffer,StringBuilder三者的使用方法和区别：<https://blog.csdn.net/qq_37856300/article/details/84340288>

Part2：笔试面试汇总：

<https://blog.csdn.net/qq_20757489/article/details/93714854>

<https://blog.csdn.net/qq_20757489/article/details/93715291>